IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for accelerating TCP (transport control protocol) processing by partitioning processing tasks between system elements in a way that scales with the elements, said method comprising:

receiving an indication on a network component that one or more packets have been received from a network;

splitting each of the one or more packets into a header and a payload and posting ~~each of the~~ header to a first buffer on a host and each payload to a second buffer on the host ~~one or more post buffers~~, using the network component;

notifying a TCP-A (transport control protocol – accelerated) driver, by the network component, that the one or more packets have arrived;

performing, using the TCP-A driver, TCP stack processing by parsing the header in at least one of the one or more packets to determine the protocol context associated with a current connection, and performing, using the TCP-A driver, TCP protocol compliance for the at least one of the one or more packets; and

performing one or more operations, using the TCP-A driver, that result in a data movement module retrieving one or more payloads of the at least one of the one or more packets from ~~an associated post~~ the second buffer and placing the one or more corresponding payloads into a read buffer wherein the data movement module comprises a DMA (direct memory access) engine.

2. (Cancelled).

3. (Previously Presented) The method of claim 1, wherein the TCP-A driver performs packet processing by processing each of the headers, the method additionally comprising fetching a next header of the one or more headers prior to completing the processing of the current header.

4. (Previously Presented) The method of claim 1, wherein said TCP-A driver performing one or more operations that result in a data movement module placing one or more corresponding payloads of the at least one of the one or more packets into a read buffer comprises sending a request to a data movement module driver to write the one or more corresponding payloads to the read buffer.

5. (Cancelled).

6. (Cancelled).

7. (Previously Presented) The method of claim 1, wherein the DMA engine resides on a chipset.

8. (Previously Presented) The method of claim 1, wherein the DMA engine resides on a host processor as a support module.

9. (Original) The method of claim 1, additionally comprising:

receiving a request on an operating system to transmit data over the network;

the operating system notifying the TCP-A driver that there is data to be transmitted;

the TCP-A driver performing one or more operations that result in the data being transmitted to the network component;

in response to receiving the data, the network component creating one or more packets for transmission by packetizing the data; and

the network component transmitting the one or more packets over the network.

10. (Currently Amended) An apparatus for accelerating TCP (transport control protocol) processing by partitioning processing tasks between elements, said apparatus comprising:

a network component configured to:

receive an indication that one or more packets have been received from a network;

split each of the one or more packets into a header and a payload and post ~~each of the~~ header to a first buffer on a host and each payload to a second buffer on the host ~~one or more post buffers~~; and

notify a TCP-A (transport control protocol – accelerated) driver that the one or more packets have arrived; and

the TCP-A driver configured to:

perform packet processing by parsing the header in at least one of the one or more packets to determine the protocol context associated with a current connection, and perform TCP protocol compliance for the at least one of the one or more packets; and

perform one or more operations that result in a data movement module retrieving one or more payloads of the at least one of the one or more packets from ~~an associated post~~ the second buffer and placing the one or more corresponding payloads into a read buffer wherein the data movement module comprises a DMA (direct memory access) engine.

11. (Previously Presented) The apparatus of claim 10, additionally comprising an operating system configured to:

receive a request to transmit data over the network; and

notify the TCP-A driver that data is ready to be transmitted;

wherein:

the TCP-A driver is capable of performing one or more operations that result in the data being transmitted to the network component; and

the network component is capable of:

creating one or more packets for transmission by packetizing the data in response to receiving the data; and

transmitting the one or more packets over the network.

12. (Cancelled).

13. (Previously Presented) The apparatus of claim 10, wherein the TCP-A driver performs packet processing by processing each of the headers, and the TCP-A driver is additionally capable of fetching a next header of the one or more headers prior to completing the processing of the current header.

14. (Currently Amended) A system for accelerating TCP (transport control protocol) processing by partitioning processing tasks between elements, said system comprising:

a chipset having a DMA (direct memory access) engine, the chipset communicatively coupled to a TCP-A (Transport Control Protocol – Accelerated) driver of a processor and to a network component;

the network component configured to:

receive an indication that one or more packets have been received from a network;

split each of the one or more packets into a header and a payload and post each ~~of the~~ header to a first buffer on a host and each payload to a second buffer on the host ~~one or more post buffers~~; and

notify the TCP-A (transport control protocol – accelerated) driver that the one or more packets have arrived; and

the TCP-A driver of the processor configured to:

perform packet processing by parsing the header in at least one of the one or more packets to determine the protocol context associated with a current connection, and perform TCP protocol compliance for the at least one of the one or more packets; and

perform one or more operations that result in a data movement module retrieving one or more payloads of the at least one of the one or more packets from ~~an associated post~~ the second buffer and placing the one or more corresponding payloads into a read buffer wherein the data movement module comprises a DMA (direct memory access) engine.

15. (Previously Presented) The system of claim 14, additionally comprising an operating system of the processor configured to:

receive a request to transmit data over the network; and

notify the TCP-A driver that data is ready to be transmitted;

wherein:

the TCP-A driver is capable of performing one or more operations that result in the data being transmitted to a network component; and

the network component is capable of:

creating one or more packets for transmission by packetizing the data in response to receiving the data; and

transmitting the one or more packets over the network.

16. (Cancelled).

17. (Previously Presented) The system of claim 14, wherein the TCP-A driver performs packet processing by processing each of the headers, and the TCP-A driver is additionally capable of fetching a next header of the one or more headers prior to completing the processing of the current header.

18. (Currently Amended) A machine-readable medium having stored thereon instructions, the instructions when executed by a machine, result in the following:

receiving an indication on a network component that one or more packets have been received from a network;

the network component splitting each of the one or more packets into a header and a payload and posting ~~each of the~~ header to a first buffer on a host and each payload to a second buffer on the host ~~one or more post buffers~~;

the network component notifying a TCP-A (transport control protocol – accelerated) driver that the one or more packets have arrived;

the TCP-A driver performing packet processing by parsing the header in at least one of the one or more packets to determine the protocol context associated with a current connection, and performing TCP protocol compliance for the at least one of the one or more packets; and

the TCP-A driver performing one or more operations that result in a data movement module retrieving one or more payloads of the at least one of the one or more packets from ~~an associated post~~ the second buffer and placing the one or more corresponding payloads into a read buffer, wherein the data movement module comprises a DMA (direct memory access) engine.

19. (Cancelled).

20. (Previously Presented) The machine-readable medium of claim 18, wherein the TCP-A driver performs packet processing by processing each of the headers, the instructions additionally result in fetching a next header of the one or more headers prior to completing the processing of the current header.

21. (Original) The machine-readable medium of claim 18, wherein the instructions that result in performing one or more operations that result in a data movement module placing one or more corresponding payloads of the at least one of the one or more packets into a read buffer additionally result in sending a request to a data movement module driver to write the one or more corresponding payloads to the read buffer.

22. (Original) The machine-readable medium of claim 18, wherein the instructions that result in said TCP-A driver performing one or more operations that result in a data movement module placing one or more corresponding payloads of the at least one of the one or more packets into a read buffer additionally result in programming the data movement module to write the one or more corresponding payloads to the read buffer.

23. (Cancelled).

24. (Previously Presented) The machine-readable medium of claim 18, wherein the DMA engine resides on a chipset.

25. (Previously Presented) The machine-readable medium of claim 18, wherein the DMA engine resides on a host processor as a support module.

26. (Original) The machine-readable medium of claim 18, the instructions additionally result in:

receiving a request on an operating system to transmit data over the network;

the operating system notifying the TCP-A driver that there is data to be transmitted;

the TCP-A driver performing one or more operations that result in the data being transmitted to the network component;

in response to receiving the data, the network component creating one or more packets for transmission by packetizing the data; and

the network component transmitting the one or more packets over the network.